Adaptive Search Center Non-linear Three Step Search

H. Y. Chung, P. Y. S. Cheung and N. H. C. Yung

Department of Electrical & Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
Tel. 2859-2685, Fax. 2559-8738, Email. nyung@eee.hku.hk

Abstract

This paper presents a new motion estimation algorithm using an adaptive search center predicted from its adjacent blocks, and a non-linear center biased search point pattern. It does not have the problem of being trapped by local minimum, and is characterized by finding the majority motion vector in one step. When compared with six other block-based search algorithms including the full-search and three-step-search, the new algorithm has an average PSNR very close to that of full-search, yet an average search time faster than the three-step-search.

1. Introduction

It is well known that motion estimation is computationally intensive, but is pivotal to the success of video coding. A good motion estimation algorithm can lead to efficient reduction of the temporal redundancies and a fast implementation, which is essential to real-time video coding applications. Among all the motion estimation techniques, block-based matching have been widely adopted by international standards such as the H.261, H.263, and MPEG.

Block matching algorithms (BMA) are based on the matching of macroblocks (MB) between two images. For each MB, a motion vector is evaluated by matching the MB within a search area, according to the Mean Absolute Error (MAE) criterion. Of all the BMA, full search (FS) [1] produces the best match. It exhaustively matches all the possible candidates within the search window, where the candidate with the least MAE would be returned as the best matched MB, and the corresponding motion vector (MV) calculated. However, FS requires massive computations which presents a significant challenge to real-time implementation. Single processor technology is known to be inadequate to provide such performance in this case. For this reason, various other search algorithms have been proposed, including three step search (TSS) [2], cross search (CS) [3], new three step search (NTSS) [4], 1D full search (1DFS) [5], one-at-a-time search (OTS) [6] and four step search (FSS) [7]. Most of these algorithms attempt to reduce the computation cost by reducing the number of search points, or varying the search pattern. The reduction in the number of search points are usually backup by the assumption that the MAE increases monotonically as the search point moves away from the global minimum. Moreover, some algorithms assume that the MV of adjacent MB in the frame are highly related. With such assumption, the number of search points is further reduced. One such algorithm is the predictive search algorithm (PSA) [8], which utilizes the linear weighting of the MV of the three adjacent MB to obtain a predicted motion vector. The sacrifice for achieving this computation reduction is the reduction in matching accuracy, and hence, the PSNR.

All the algorithms mentioned above have been reported to be able to reduce the computational requirement significantly. Among them, TSS has been the most popular one due to its simplicity. However, TSS suffers from two problems: first, its PSNR is substantially lower than that of FS and second, it can be easily trapped in a non-optimum solution. Based on these observations, this research was motivated to develop an algorithm based on TSS, but without the said problems.

In this paper, we propose an algorithm called the adaptive center non-linear three step search (ACNTSS), which is a modified version of TSS, but works even faster, produces PSNR closer to that of FS and would not be trapped in a non-optimum solution. Essentially, it differs from TSS in three aspects: first, its search center is predicted from the left and top adjacent MB based on a model we derived. Second, it employs a non-linear center biased search point pattern which spirals outward. Third, it stops sooner when the minimum position is closer to the search center. Four sequences 'Susie', 'Football', 'Flower Garden' and 'Mobile & Calendar' were used to test the proposed algorithm, together with FS, TSS, FSS, 1DFS, CS and NTSS for comparison. From our simulation results, we observed that the search time of the ACNTSS algorithm is 89%, 82%, 62% and 45% that of TSS for these sequences, respectively. In terms of PSNR, ACNTSS is 0.151 dB, 0.025 dB, 0.129 dB and 0.041 dB lower than that of FS for the four sequences, which is

0-8186-8821-1/98 $10.00

0.362 dB, 0.118 dB, 0.598 dB and 0.773 dB higher than that of TSS. Besides, ACNTSS is found to have a PSNR performance very close to that of FS. From our simulation results, ACNTSS is able to achieve over 99% of the PSNR of that of FS for all four sequences.

2. ACNTSS Algorithm

2.1. Search Center Prediction

Assume the motion of an object are mainly translational and rotational, which can be further generalized as rotational motion about a center (at infinity for translational motion). Suppose an object in the current frame spans through several blocks of W×W each as shown in Fig.1, the relationship between the MV of the adjacent blocks can be deduced using the following model.

Let MV of block A be $\mathbf{A} = a_x\mathbf{i} + a_y\mathbf{j}$,
MV of block B be $\mathbf{B} = b_x\mathbf{i} + b_y\mathbf{j}$ and
MV of block T be $\mathbf{T} = t_x\mathbf{i} + t_y\mathbf{j}$.

As MV can be regarded as the velocity of the object in pixels displacement per unit time between the current and reference frame, the velocity components of **T** in the directions of **i** and **j** are simply $t_x = a_x$ and $t_y = b_y$. If the object spans across blocks A, B and T, the target MV, **T**, can be predicted from the MV of the two adjacent blocks A and B. Before doing this, we need to determine whether the three blocks lie on the same object, as shown below.

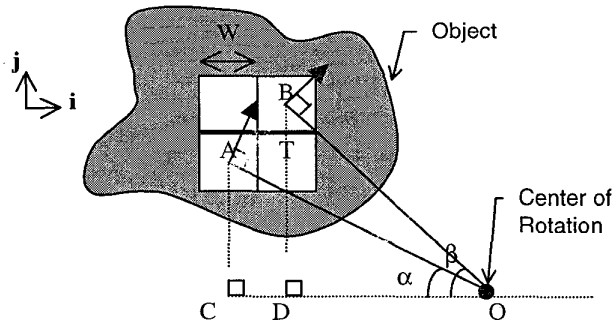

Fig.1 Rotational motion of object

If A and B lie on the same object, block A and B should have the same angular velocity about the center of rotation. Therefore, if the angular velocity ω_a of block A and angular velocity ω_b of block B are equal, we can say that they lie on the same object. To prove that this indeed is the case, we have

$$OD = OA\cos\alpha - W = OB\cos\beta \quad (1)$$

$$BD = OA\sin\alpha + W = OB\sin\beta \quad (2)$$

Solving: (1) & (2)

$$OB = \frac{W(\sin\alpha+\cos\alpha)}{\sin(\beta-\alpha)} \quad , \quad OA = \frac{W(\sin\beta+\cos\beta)}{\sin(\beta-\alpha)} \quad (3)$$

And since

$$\boldsymbol{B}\times\boldsymbol{A} = \|\boldsymbol{B}\|\|\boldsymbol{A}\|\sin(\beta-\alpha)\mathbf{k}, \text{ where } \mathbf{k} = \mathbf{i}\times\mathbf{j} \quad (4)$$

Therefore,

$$\|\omega_a\| = \frac{\|\mathbf{A}\|}{OA} = \frac{(\mathbf{B}\times\mathbf{A})\cdot\mathbf{k}}{W\|\mathbf{B}\|(\sin\beta+\cos\beta)} = \frac{(\mathbf{B}\times\mathbf{A})\cdot\mathbf{k}}{W(b_x+b_y)} \quad (5)$$

$$\|\omega_b\| = \frac{\|\mathbf{B}\|}{OB} = \frac{(\mathbf{B}\times\mathbf{A})\cdot\mathbf{k}}{W\|\mathbf{A}\|(\sin\alpha+\cos\alpha)} = \frac{(\mathbf{B}\times\mathbf{A})\cdot\mathbf{k}}{W(a_x+a_y)} \quad (6)$$

Thus, for $a_x + a_y \neq 0$ and $b_x + b_y \neq 0$

$$\|\omega_a\| = \|\omega_b\| \quad \text{iff} \quad a_x + a_y = b_x + b_y \quad (7)$$

If (7) is satisfied, it is highly likely that block T also lies on the same object. Therefore, **T** can be predicted from **A** and **B**, as illustrated below, which is used as search center in the proposed algorithm:

$$\mathbf{T} = a_x\mathbf{i} + b_y\mathbf{j} \quad (8)$$

2.2 Search Center Biased Search Pattern

The distribution of MV in image sequence with gentle and smooth motion is highly biased towards the central region. As discussed in [4,8], nearly 83% of the MV of 'Miss America' and 81% of the MV of 'Tennis' are enclosed in the central 5x5 region. Similar results have been obtained in the 'Susie' and 'Football' sequences. With this property, it is reasonable to place more search points in the center region of the search window to get more 'samples'. As the search center can now be predicted using the left and top adjacent MB, it is therefore reasonable to place more search points near the predicted search center and stop the search when the minimum position is close to the search center.

The algorithm works like this: It first tests whether the search center can be predicted by using Eqt.(7). If not, the first step would consider the search points:

(0, 0), (±*step*,0), (0, ±*step*), (±*step*, ±*step*)

for *step* = 1, 2, 4, 8 ....., as far as the points are within the search window. Fig.2 shows the distribution of search points when the search center cannot be predicted.

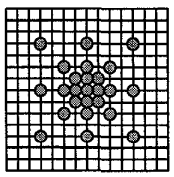

Fig.2. Unpredicted search center

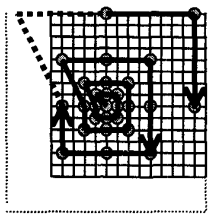

Fig. 3. Predicted search center

However, if the search center can be predicted, the first step would consists of the search points:

(i_c, j_c), $(i_c \pm step, j_c)$, $(i_c, j_c \pm step)$, $(i_c \pm step, j_c \pm step)$

where (i_c, j_c) is the predicted search center, for $step$ = 1, 2, 4, 8 .....,

as far as the points are within the search window. The algorithm then visits the search points in an outward spiral manner as shown in Fig.3. Whenever the current minimum position is found within the spiral but not the boundary, the first step search completes.

If the minimum position found in the first step is the search center or its eight neighbors, the search is stopped in the first step and this is called **first-step-stop**. If the minimum position is one of the corner neighbors of the search center, two more search points are visited before the search completes. It is depicted in Fig. 4.

If the minimum position found in the first step is not the search center or its eight neighbors, other step search would be carried out to refine the minimum position found. The refinement process is the same as that in TSS but the refinement process takes a smaller step size and hence the search stops sooner if the minimum position found in the first step search is closer to the search center. Fig.5 and Fig.6 depict the refine process.

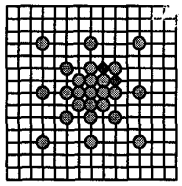

Fig.4. First-step-stop

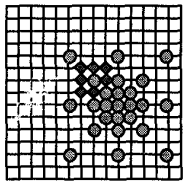

Fig.5. Refinement 1

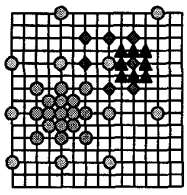

Fig. 6. Refinement 2

3. Simulation Results

The ACNTSS algorithm was simulated using four 90 frames MPEG test sequences 'Football', 'Susie', 'Flower Garden' and 'Mobile & Calendar'. All the sequences are encoded into MPEG2 bitstreams using the MPEG2 encoder from MSSG[10]. Each group of pictures contains 15 frames. The block size is 16 × 16 and a 2-frame interpolation structure was used. The search range is -31 to 31 pixels for 'Football and 'Susie' and -15 to 15 for 'Flower Garden' and 'Mobile & Calendar'. Other fast BMA such as TSS, NTSS, FSS, CS, 1DFS have also been implemented for comparing their performance with ACNTSS. The performance is evaluated on two counts: Peak-to-peak SNR (PSNR) and the average search times per block. These results are shown in Table 1 and Table 2.

Table 1. Performance of six block matching algorithms for 'Football', 'Susie' (*search time)

Sequence	'Football'		'Susie'	
BMA	ST*	PSNR	ST	PSNR
FS	3969	16.449	3969	22.626
ACNTSS	36.29	16.298	33.79	22.601
FSS	26.88	16.249	27.29	22.505
NTSS	30.5	16.091	29	22.398
1DFS	186	16.021	186	22.404
TSS	41	15.936	41	22.413
CS	25	15.367	25	21.839

Table 2. Performance of six block matching algorithms for 'Flower Garden' & 'Mobile & Calendar'

Sequence	'Flower Garden'		'Mobile & Calendar'	
BMA	ST	PSNR	ST	PSNR
FS	961	13.666	961	13.230
ACNTSS	20.57	13.537	14.74	13.189
FSS	21.78	12.752	17.96	13.176
NTSS	28.83	12.786	19.26	13.137
1DFS	90	13.660	90	13.188
TSS	33	12.939	33	12.416
CS	21	10.682	21	11.598

From Table 1, FS is the slowest but it also gives the best PSNR. On the other extreme, the fastest BMA is CS, but is also the poorest in PSNR. Although ACNTSS is not the fastest BMA as shown, it has the best PSNR performance besides FS. From Table 2, ACNTSS is the fastest BMA algorithm apart from FS. Although 1DFS have better PSNR than ACNTSS for the 'Flower Garden' sequence, it searches much longer than ACNTSS (90 compared with 20.57).

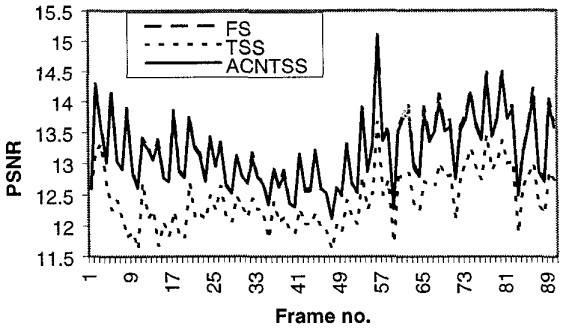

Fig.7. PSNR for 'Mobile & Calendar'

Fig. 7 shows the PSNR comparison between TSS and ACNTSS for the 'Mobile & Calendar' sequence. As shown, the PSNR of ACNTSS is very close to that of FS and much better than that of TSS. Fig.8 depicts a method for evaluating the overall performance of the BMAs, with the PSNR on the Y-axis normalized to the PSNR of FS and the average search time per block on the X-axis normalized to the search time of FS.

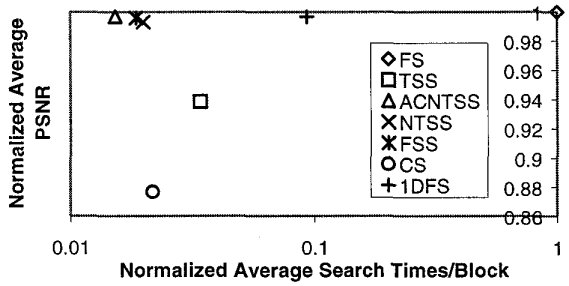

Fig. 8. Overall quality for 'Mobile & Calendar'

Here, we are interested in the relative performance rather than the absolute. On the metric shown in Fig. 8, the best algorithms would lie on the top left hand corner, while the worst algorithm would lie on the bottom right hand corner. In this case, the ACNTSS algorithm is the closest to the top than the other algorithms, indicating its better overall quality.

4. Conclusions

In conclusion, we have proposed a new algorithm that is built around a search center prediction model and a non-linear search pattern biased towards the search center. The prediction is adaptive, depending on whether the left and top neighbors are on the same moving object. Moreover, it offers a mathematical approach to finding the search center. Our results show that this new algorithm is faster than TSS, but with PSNR very close to FS. Future direction will be focused on increasing the first-step-stop and improving the overall PSNR.

5. References

[1] ISO/IEC 13818-2 Coding of moving picture and associated audio, 1995.

[2] T. Koga, el al, "Motion-compensated inter-frame coding for video conferencing," in *Proc. NTC81*, New Orleans, LA, Nov. 1981, pp.C9.6.1-9.6.5

[3] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol.38, no.7, July 1990, pp.950-953.

[4] R. Li, B. Zeng, and M. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. CASVT*, vol. 4, no. 4, Aug. 1994, pp.438-442.

[5] M. J. Chen, L. G. Chen and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. CASVT*, vol. 4, no. 5, Oct. 1994, pp.504-509.

[6] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *ICC'84*, 1984, pp.521-526

[7] M. P. Lai and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. CASVT*, vol. 6, no. 3, Jun. 1996, pp.313-317.

[8] L. Luo, C. Zou, X. Gao, "A new prediction search algorithm for block motion estimation in video coding," *IEEE Trans. Consumer Electronics*, vol. 43, no. 1, 1997, pp.56-61.

[9] Bede Liu and Andre Zaccarin, "New fast algorithms for estimation of block motion vectors", *IEEE Trans. CASVT*, Apr. 1993, vol. 3, no.2, pp.148-157.

[10] MPEG2 Encoder v1.2 from MPEG Software Simulation Group.